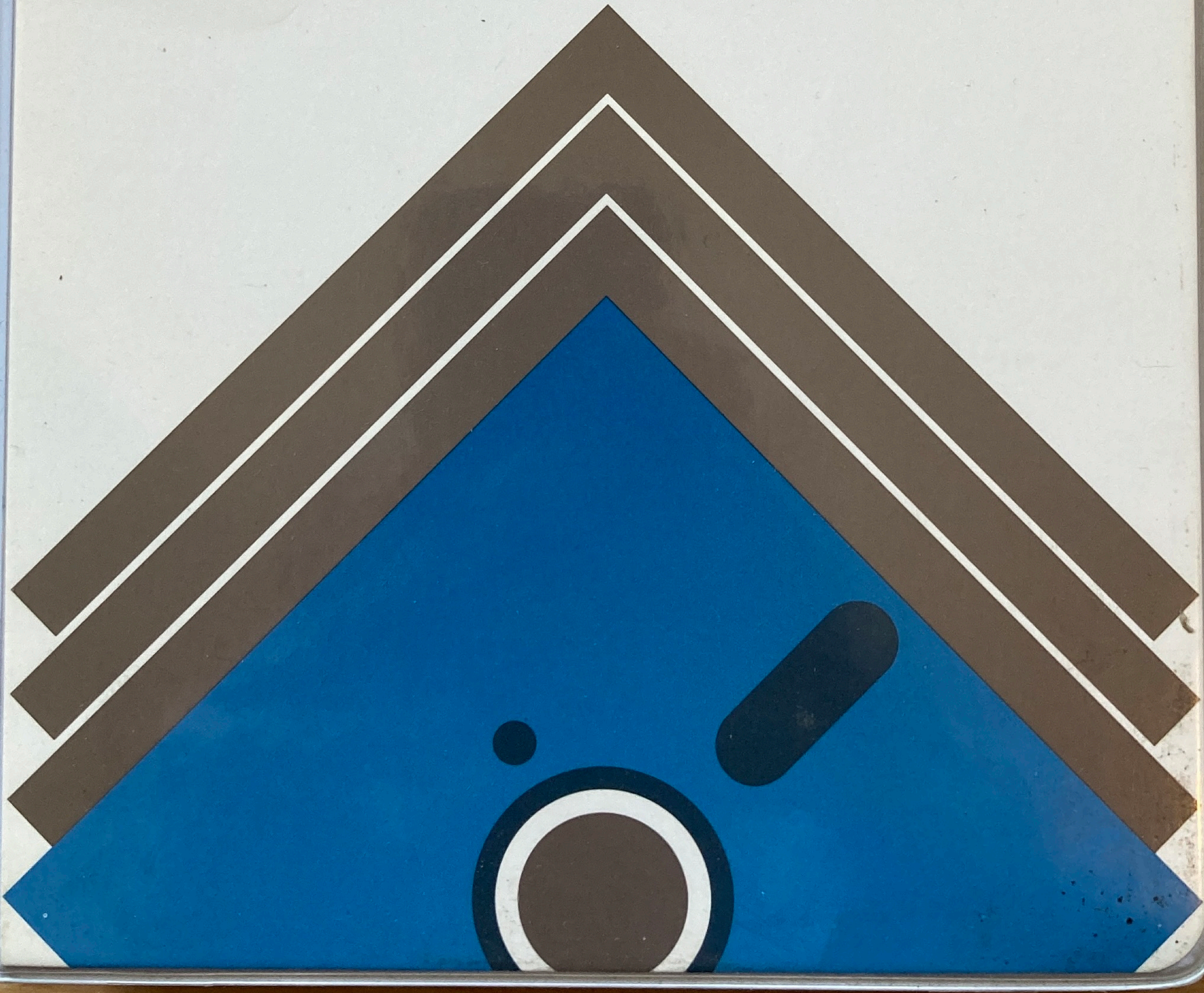
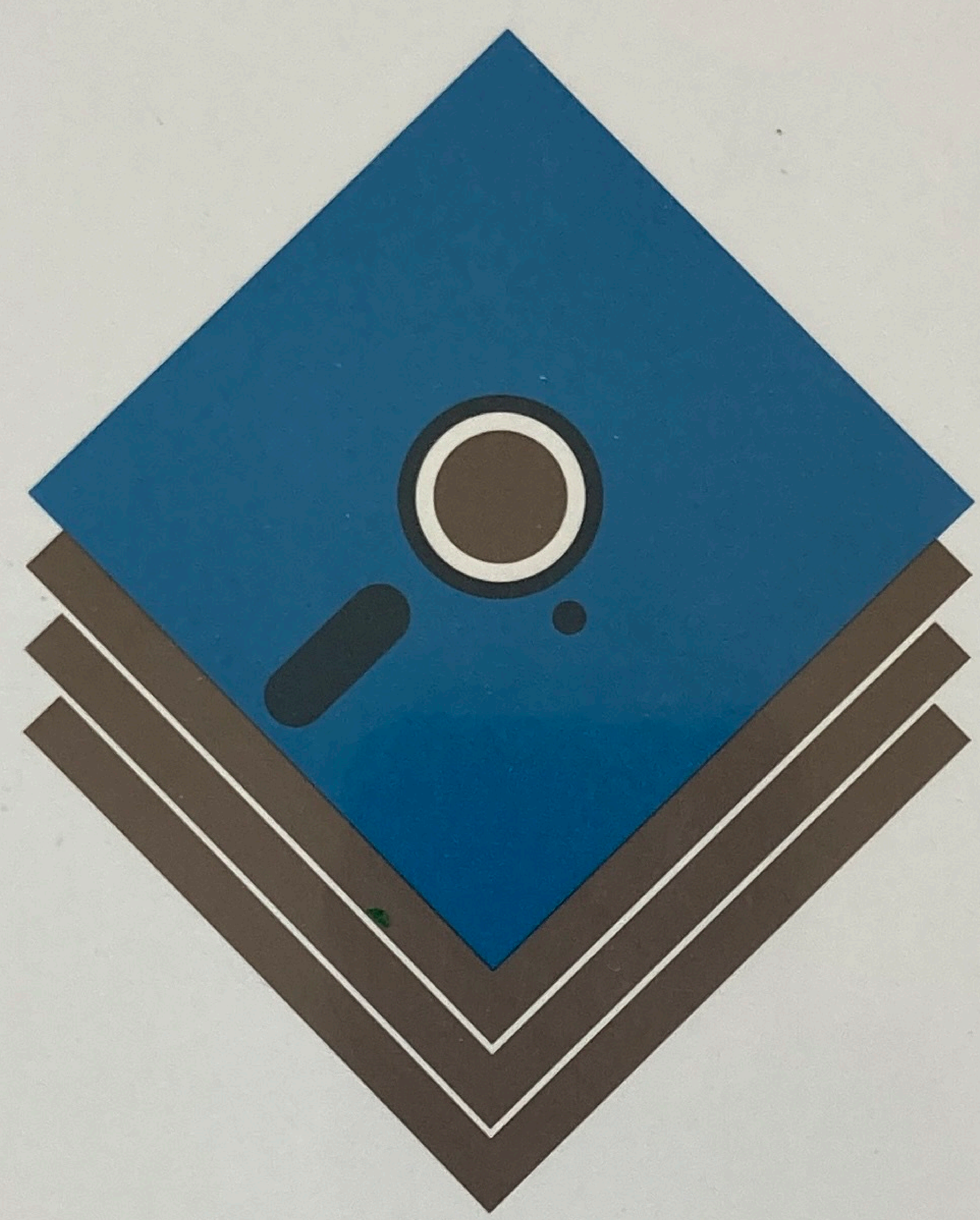
COMPress A Division of Van Nostrand Reinhold Company, Inc.

EnBASIC™
AUTHORING SYSTEM

FOR THE APPLE II PLUS*, //e*
& FRANKLIN ACE 1000**

Paul Tenczar, Stanley Smith and Allen Avner





COMPress
A Division of Van Nostrand Reinhold Company, Inc.

™ A trademark of Computer Teaching Corporation
* Registered trademark of Apple Computer Company
** Registered trademark of Franklin Computer Corporation

ISBN 0-933694-02-4

EnBASIC™
AUTHORING SYSTEM
FOR THE APPLE II PLUS®, //e®
& FRANKLIN ACE 1000™
Paul Tenczar, Stanley Smith and Allen Avner

EnBASIC™
AUTHORING SYSTEM
FOR THE APPLE II PLUS*, //e*
& FRANKLIN ACE 1000*
Paul Tenczar, Stanley Smith and Allen Avner
COMPress A Division of Van Nostrand Reinhold Company Inc.
EnBASIC™ AUTHORING SYSTEM

COMPress
A Division of Van Nostrand Reinhold Company, Inc.
*® trademark of Computer Teaching Corporation
**Registered trademark of Apple Computer Company
***Registered trademarks of Franklin Computer Corporation
ISBN 0-933694-02-4
AUTHORING SYSTEM

EnBASIC Master
Tenczar, Smith & Avner
DOS 3.3

COMPress, A Div. Van Nostrand Reinhold
P.O. Box 102 Wentworth, NH 03282

COMPress A Division of Van Nostrand Reinhold Company, Inc.

EnBASIC™
AUTHORING SYSTEM

FOR THE APPLE II PLUS*, //e*
& FRANKLIN ACE 1000**

Paul Tenczar, Stanley Smith and Allen Avner

EnBASIC Duplicate
Tenczar, Smith & Avner DOS 3.3

COMPress, A Div. of Van Nostrand Reinhold
P.O. Box 102 Wentworth, NH 03282

EnBASIC Demo
Tenczar, Smith & Avner DOS 3.3

COMPress, A Div. of Van Nostrand Reinhold
P.O. Box 102 Wentworth, NH 03282

EnBASIC Master
Tenczar, Smith & Avner
DOS 3.3

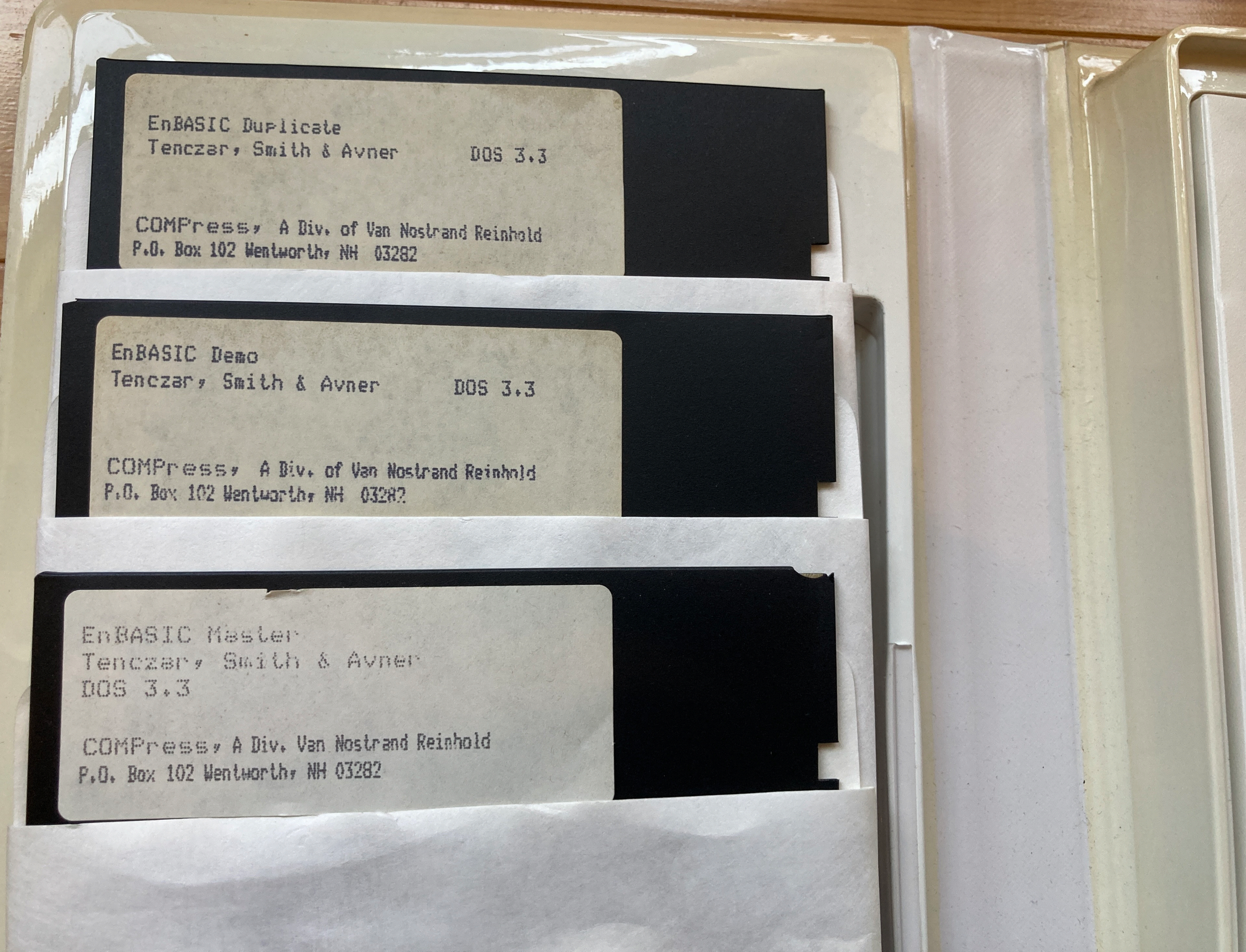
COMPress, A Div. Van Nostrand Reinhold
P.O. Box 102 Wentworth, NH 03282

COMPress A Division of Van Nostrand Reinhold Company, Inc.

EnBASIC™
AUTHORING SYSTEM

FOR THE APPLE II PLUS*, //e*
& FRANKLIN ACE 1000**

Paul Tenczar, Stanley Smith and Allen Avner



EnBASIC Duplicate
Tenczar, Smith & Avner DOS 3.3

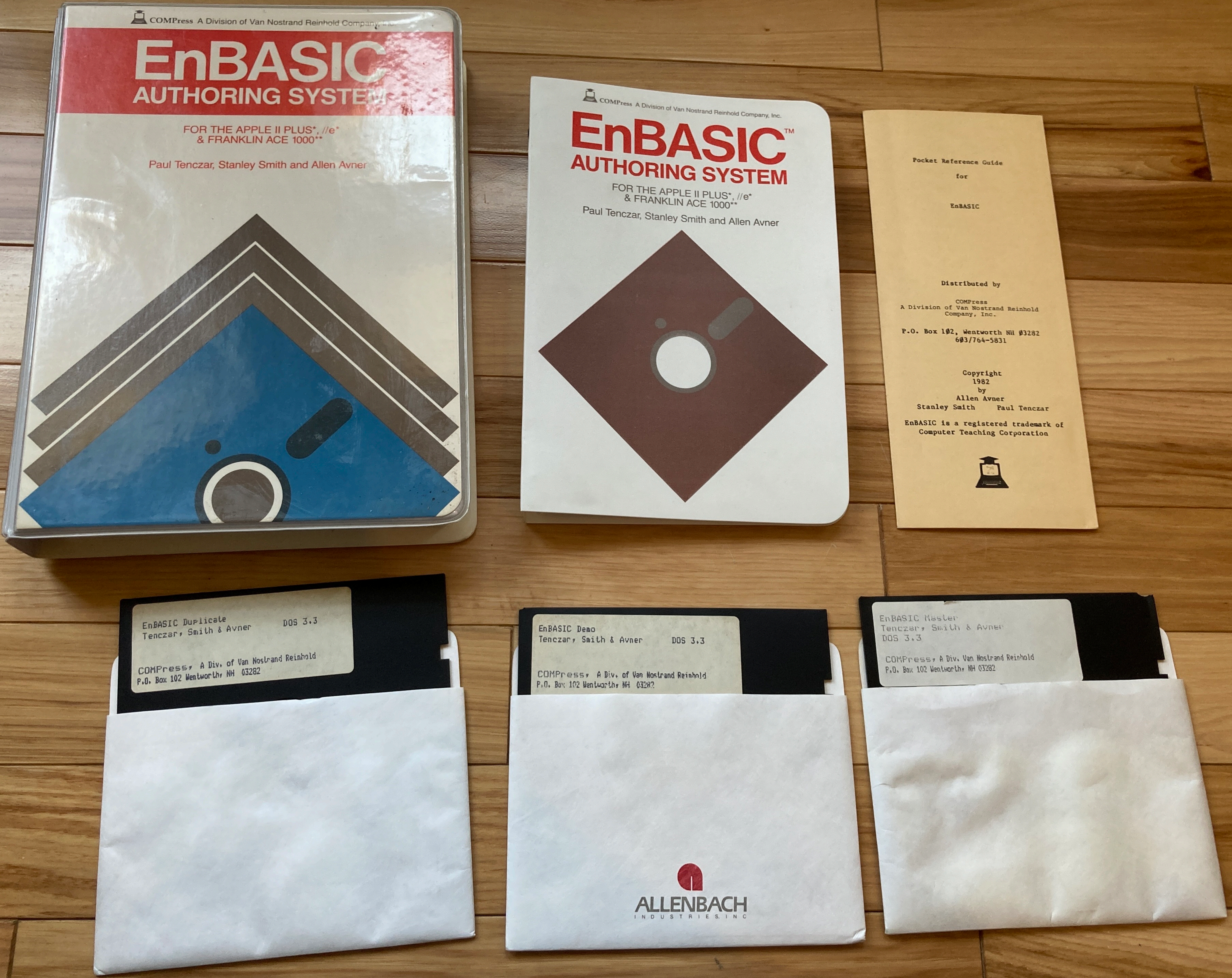
COMPress, A Div. of Van Nostrand Reinhold
P.O. Box 102 Wentworth, NH 03282

EnBASIC Demo
Tenczar, Smith & Avner DOS 3.3

COMPress, A Div. of Van Nostrand Reinhold
P.O. Box 102 Wentworth, NH 03282

EnBASIC Master
Tenczar, Smith & Avner
DOS 3.3

COMPress, A Div. Van Nostrand Reinhold
P.O. Box 102 Wentworth, NH 03282



COMPress A Division of Van Nostrand Reinhold Company, Inc.

EnBASIC
AUTHORING SYSTEM

FOR THE APPLE II PLUS*, //e*
& FRANKLIN ACE 1000**

Paul Tenczar, Stanley Smith and Allen Avner

COMPress A Division of Van Nostrand Reinhold Company, Inc.

EnBASIC™
AUTHORING SYSTEM

FOR THE APPLE II PLUS*, //e*
& FRANKLIN ACE 1000**

Paul Tenczar, Stanley Smith and Allen Avner

Pocket Reference Guide

for

EnBASIC

Distributed by

COMPress
A Division of Van Nostrand Reinhold
Company, Inc.

P.O. Box 102, Wentworth NH 03282
603/764-5831

Copyright
1982
by
Allen Avner
Stanley Smith   Paul Tenczar

EnBASIC is a registered trademark of
Computer Teaching Corporation

EnBASIC Duplicate   DOS 3.3
Tenczar, Smith & Avner

COMPress, A Div. of Van Nostrand Reinhold
P.O. Box 102 Wentworth, NH 03282

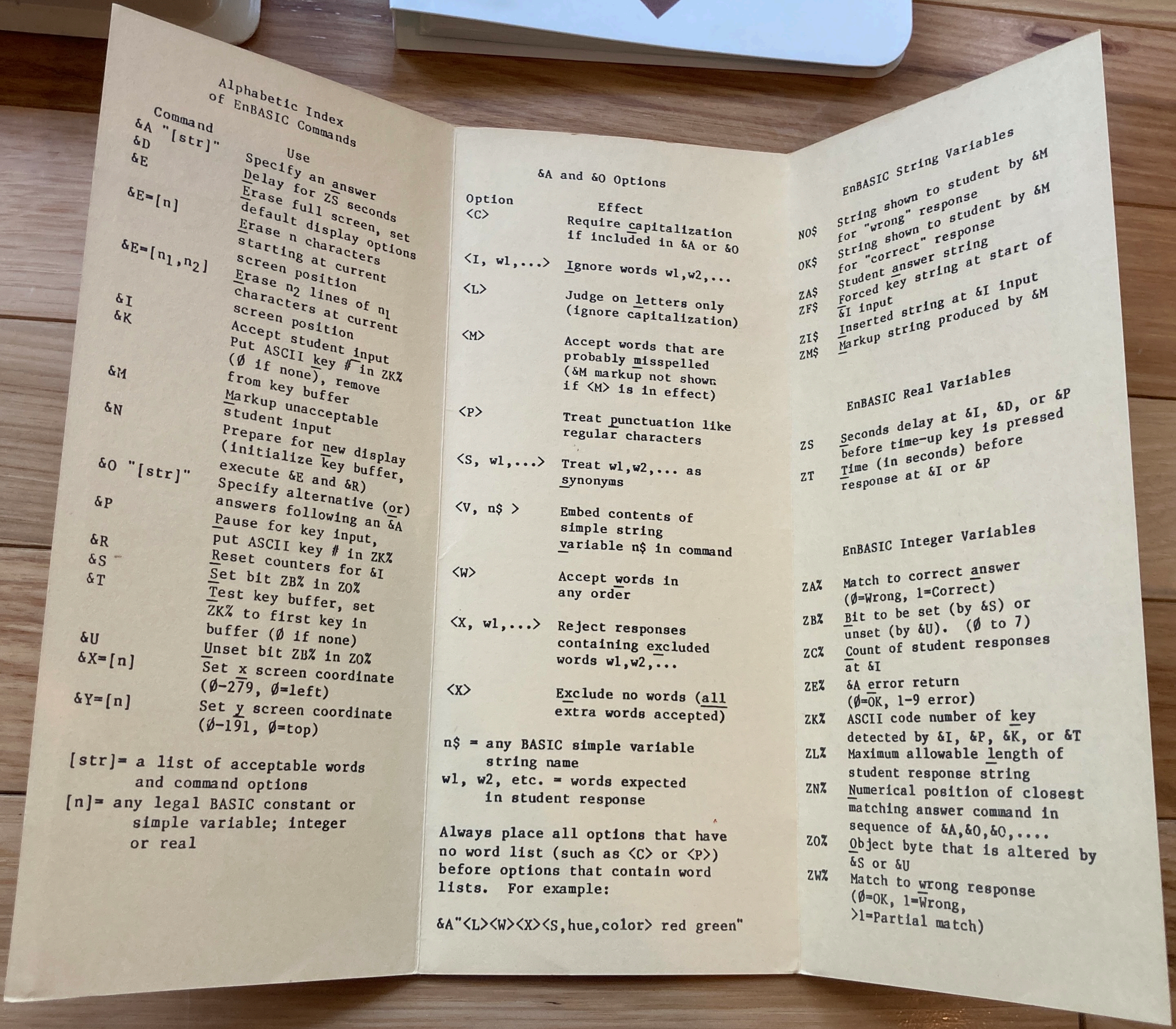
EnBASIC Demo
Tenczar, Smith & Avner   DOS 3.3

COMPress, A Div. of Van Nostrand Reinhold
P.O. Box 102 Wentworth, NH 03282

ALLENBACH
INDUSTRIES, INC

EnBASIC Master
Tenczar, Smith & Avner
DOS 3.3

COMPress, A Div. Van Nostrand Reinhold
P.O. Box 102 Wentworth, NH 03282



Alphabetic Index of EnBASIC Commands

Command	Use
&A "[str]"	Specify an answer
&D	Delay for ZS seconds
&E	Erase full screen, set default display options
&E=[n]	Erase n characters starting at current screen position
&E=[n_1,n_2]	Erase n_2 lines of n_1 characters at current screen position
&I	Accept student input
&K	Put ASCII key # in ZK% (0 if none), remove from key buffer
&M	Markup unacceptable student input
&N	Prepare for new display (initialize key buffer, execute &E and &R)
&O "[str]"	Specify alternative (or) answers following an &A
&P	Pause for key input, put ASCII key # in ZK%
&R	Reset counters for &I
&S	Set bit ZB% in ZO%
&T	Test key buffer, set ZK% to first key in buffer (0 if none)
&U	Unset bit ZB% in ZO%
&X=[n]	Set x screen coordinate (0-279, 0=left)
&Y=[n]	Set y screen coordinate (0-191, 0=top)

[str]= a list of acceptable words and command options

[n]= any legal BASIC constant or simple variable; integer or real

&A and &O Options

Option	Effect
<C>	Require capitalization if included in &A or &O
<I, w1,...>	Ignore words w1,w2,...
<L>	Judge on letters only (ignore capitalization)
<M>	Accept words that are probably misspelled (&M markup not shown if <M> is in effect)
<P>	Treat punctuation like regular characters
<S, w1,...>	Treat w1,w2,... as synonyms
<V, n$ >	Embed contents of simple string variable n$ in command
<W>	Accept words in any order
<X, w1,...>	Reject responses containing excluded words w1,w2,...
<X>	Exclude no words (all extra words accepted)

n$ = any BASIC simple variable string name

w1, w2, etc. = words expected in student response

Always place all options that have no word list (such as <C> or <P>) before options that contain word lists. For example:

```
&A"<L><W><X><S,hue,color> red green"
```

EnBASIC String Variables

NO$	String shown to student by &M for "wrong" response
OK$	String shown to student by &M for "correct" response
ZA$	Student answer string
ZF$	Forced key string at start of &I input
ZI$	Inserted string at &I input
ZM$	Markup string produced by &M

EnBASIC Real Variables

ZS	Seconds delay at &I, &D, or &P before time-up key is pressed
ZT	Time (in seconds) before response at &I or &P

EnBASIC Integer Variables

ZA%	Match to correct answer (0=Wrong, 1=Correct)
ZB%	Bit to be set (by &S) or unset (by &U). (0 to 7)
ZC%	Count of student responses at &I
ZE%	&A error return (0=OK, 1-9 error)
ZK%	ASCII code number of key detected by &I, &P, &K, or &T
ZL%	Maximum allowable length of student response string
ZN%	Numerical position of closest matching answer command in sequence of &A,&O,&O,....
ZO%	Object byte that is altered by &S or &U
ZW%	Match to wrong response (0=OK, 1=Wrong, >1=Partial match)

Error Codes for &A and &O Commands
(in ZE% after &A or &O executed)

ZE%	Indicates
0	No errors
1	Student input string too long (>250 characters)
2	Author input string too long (>250 characters)
3	Quote mark missing at start of &A or &O
4	No comma or space between V and BASIC variable: <VC$> should be <V,C$>
5	No > at end of option: <V,C$ should be <V,C$>
6	Bad string variable name
7	Too many words in student response (>32)
8	Too many total student and author words (student words x author words>224)
9	<C>,<M>,<P>,<U> not before author's words

Useful PEEKS and POKES

BASIC program space left (in bytes):
16384 - PEEK(175) - 256*PEEK(176)

Screen location of next output:
X = PEEK(224) + 256*PEEK(225)
Y = PEEK(226)

EnBASIC special display functions
On=POKE 24590,0   Off=POKE 24590,1

EnBASIC Pointer Addresses

Normal Display Table	24580,24581
Alternate Display Table	24582,24583
Key Table	24584,24585
& Continue Address	24586,24587
First loc. above EnBASIC	24588,24589

Useful "Standard Table" Effects

FONT SELECTION

Standard-size letters	@1
Double-size letters	@2
Alternate Font	@F
Standard Font	@S

CHARACTER POSITIONING

Fixed spacing	@0
Proportional spacing	@P
Backspace	@B
Carriage Return (1 line)	@R
Carriage Return (1.5 line)	CTRL-C

MARGIN CONTROL

Set new margin	@CTRL-P
Automatic CR at right margin	@CTRL-R
Wrap-around after right margin	@CTRL-Q

UPPER/LOWER CASE CONTROL

Show all letters as upper case	@H
Show all letters as lower case	@L
Show letters as actually coded	@Z
Toggle upper/lower case	@@

SUPERSCRIPTS AND SUBSCRIPTS

Superscript	@U
Subscript	@D
Locking superscript	@+
Locking subscript	@;

SPECIAL DISPLAY MODES

ERASE mode	@!
OVERSTRIKE mode	@#
REWRITE mode	@$
EXCLUSIVE-OR mode	@%
INVERSE mode	@&
NON-PLOT mode	@*

Pocket Reference Guide

for

EnBASIC

Distributed by

COMPress
A Division of Van Nostrand Reinhold
Company, Inc.

P.O. Box 102, Wentworth NH 03282
603/764-5831

Copyright
1982
by
Allen Avner
Stanley Smith   Paul Tenczar

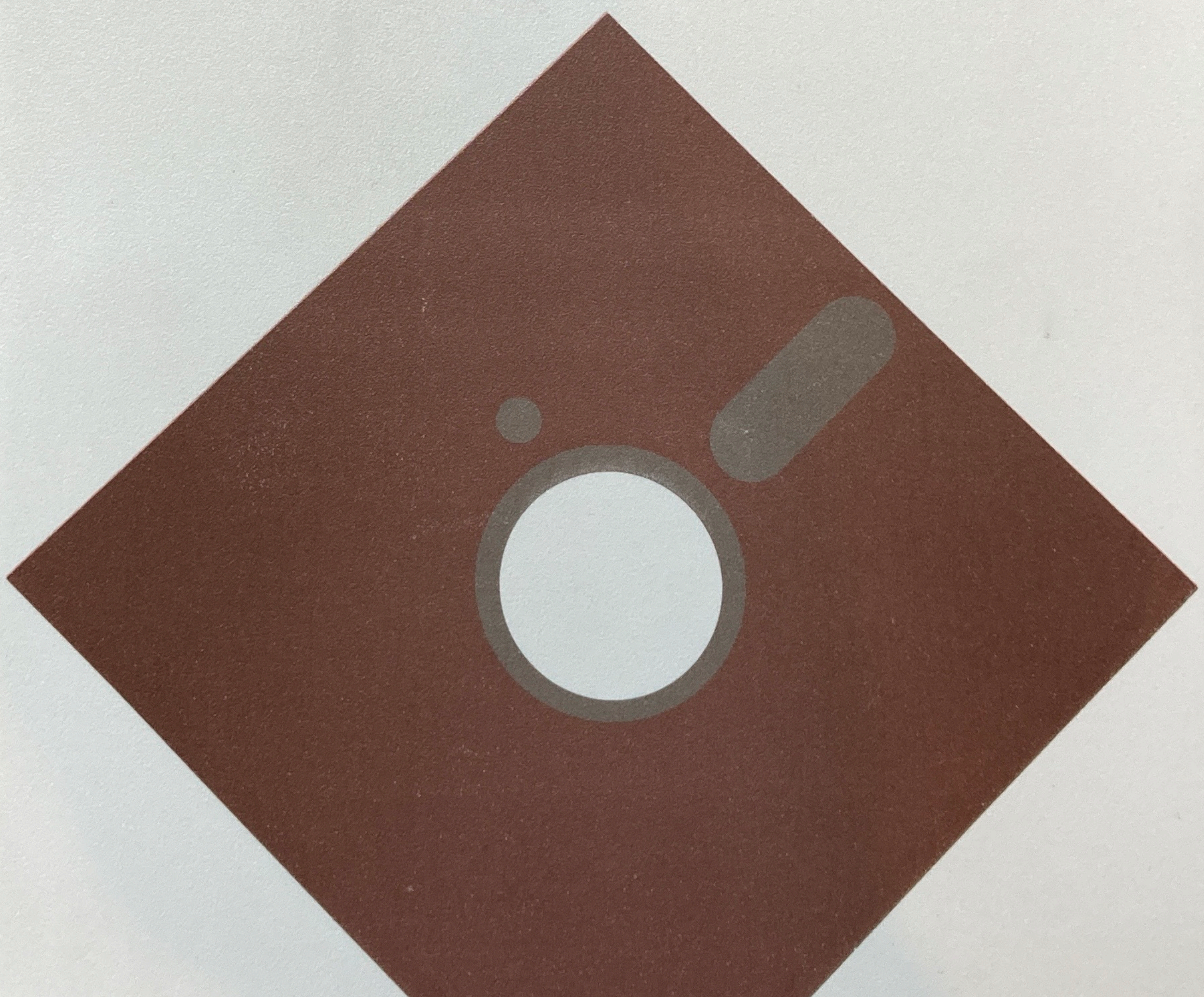
EnBASIC is a registered trademark of
Computer Teaching Corporation

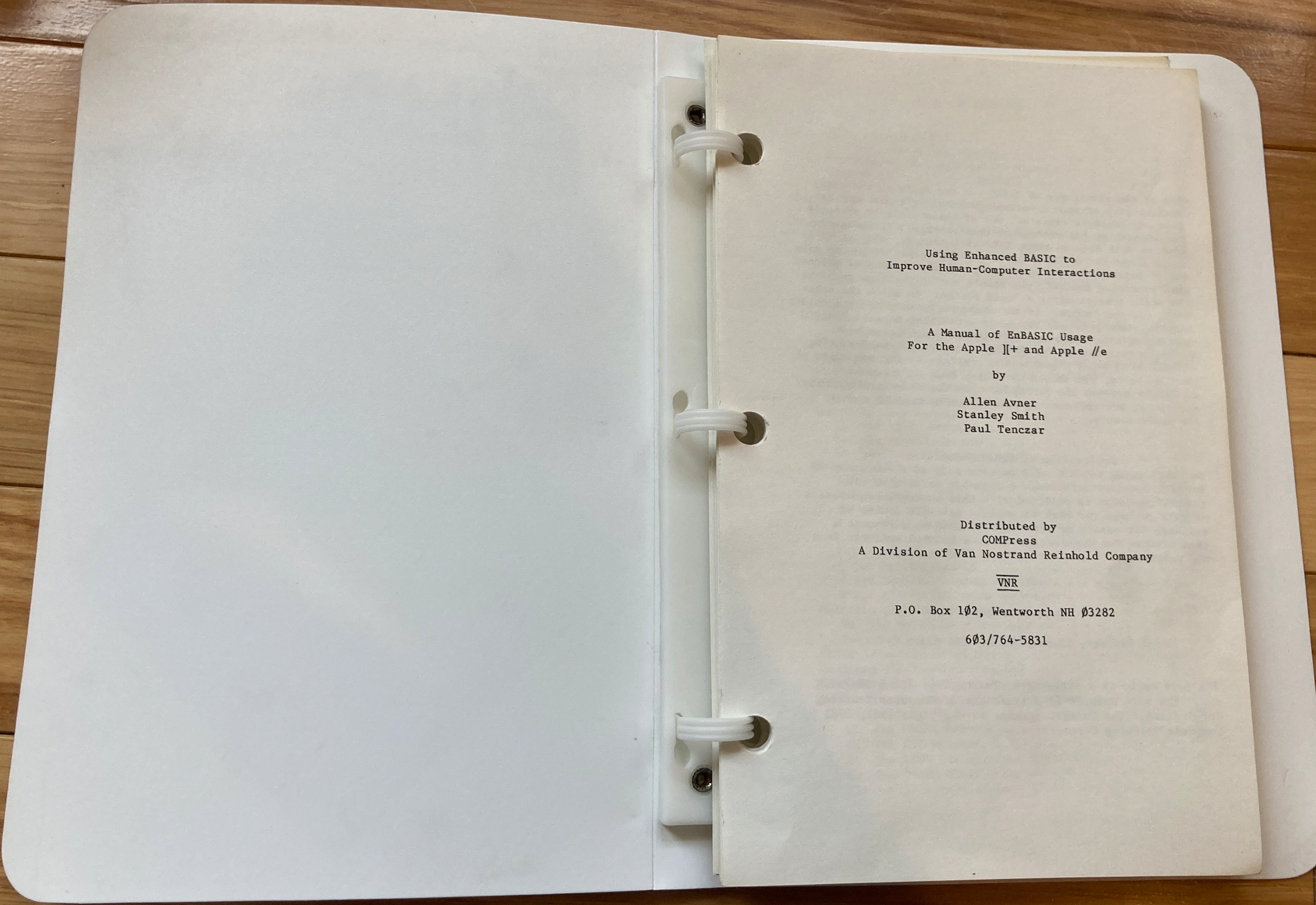
COMPress A Division of Van Nostrand Reinhold Company, Inc.

EnBASIC™
AUTHORING SYSTEM

FOR THE APPLE II PLUS*, //e*
& FRANKLIN ACE 1000**

Paul Tenczar, Stanley Smith and Allen Avner





Using Enhanced BASIC to Improve Human-Computer Interactions

A Manual of EnBASIC Usage
For the Apple][+ and Apple //e

by

Allen Avner
Stanley Smith
Paul Tenczar

Distributed by
COMPress
A Division of Van Nostrand Reinhold Company

VNR

P.O. Box 102, Wentworth NH 03282

603/764-5831

Here is a working example. Press CTRL-RESET and type it in (or copy program EX IV-6 from your Demo diskette).

```
Program                          Comments
NEW
110  &N                          Prepare for New interaction
120  PRINT"WHAT GOES 'MEOW'?"
130  &I                          Input  response
140  &A"<S,@LCAT,FELINE>"        Answer command with synonyms
150  &M                          Mark up the answer
160  IF ZA% THEN 300             If OK, go to 300
170  ZK%=0                       Zero ZK% as a flag
180  HTAB 10: VTAB 10            Location of help messages
190  IF ZC%=1 THEN PRINT"@LBEGINS WITH 'C'"
200  IF ZC%>1 THEN PRINT"@LTRY 'CAT'"
210  IF ZK% THEN 130             Return to input on second pass
220  &T: IF NOT ZK% THEN 220     Loop until keypress
230  GOTO 180                    Keypress--erase help message
300  HTAB 10: VTAB 10: PRINT "GOOD!"
RUN
```

Lines 110 to 150 set up the question, allow input, specify the answer, and provide the markup. Line 160 tests to see if the answer is right (ZA%= 1) and, if it is, gives the message "GOOD!" and moves on beyond the question.

If the answer is not right, ZK% is zeroed as a flag on line 170. Line 180 positions help messages on the screen. On the first attempt to answer the question, the BASIC variable ZC% is set to 1 by the answer command; thus, the hint on line 190 is printed on the screen. More than one try gives the hint on line 200. Next, line 210 checks the value of ZK%. Since ZK% was set to zero on line 170, the test fails and the program goes on to 220.

The command, &T, on line 220 Tests the keyboard to see if the student has typed a key. If no key has been pressed, ZK% is not altered from the value of 0 set on line 170. As long as no key is pressed, looping continues at line 220. Once any key is pressed, ZK% is set to a non-zero value, and control falls through to line 230 and thence to line 180 and once again through the help message. Since EnBASIC, by default, writes in exclusive-or mode, rewriting the help message over itself results in automatic erasing of the old message (a very handy feature of exclusive-or mode!). We now have a non-zero value in ZK%, so this time when we come to line 210, we branch back to the &I on line 130 where the key pressed at &T is acted on. Thus, a student can type "felin", press RETURN, receive feedback about the missing "e", type that letter, and immediately see the corrected word, ready for rejudging. Such approaches do much to make interaction with the computer a friendly experience.

Key-Buffering and Single-Key Sensing

Student responses made up of natural-language strings in a well-defined context are one important aspect of advanced CAI. The &I, &A, and &O commands provide tools for handling such reponses in BASIC.

Another vital aspect of advanced CAI consists of real-time user interaction with a computer that is busy updating simulations or graphics. Designing such CAI environments requires methods of controlling access to the keyset and sensing individual keys both reliably (keys must not be lost or misinterpreted) and efficiently (key processing should not degrade simulations). The commands described in this section provide a variety of methods for meeting these needs. Among these commands are the &P and &T commands which have already been used in some of our examples.

Key Buffering

EnBASIC makes use of a multi-key input buffer to prevent loss of keys entered during simulations or graphics processing. The various pause, delay, and input commands treat keys in this buffer differently. All but &T remove the oldest key from the buffer. &T tests for keys but does not alter the buffer -- any keys in the buffer remain there. All EnBASIC commands place keys from the keyset into this multi-key buffer. However, keys entered during long, complex non-EnBASIC computations do risk being lost. To prevent such loss, &T commands can be placed in these parts of the program. Keyset entries are then stored in the key buffer for later detection by the program. Execution of the &N command clears the multi-key buffer to prevent unintended interaction between sections of the lesson.

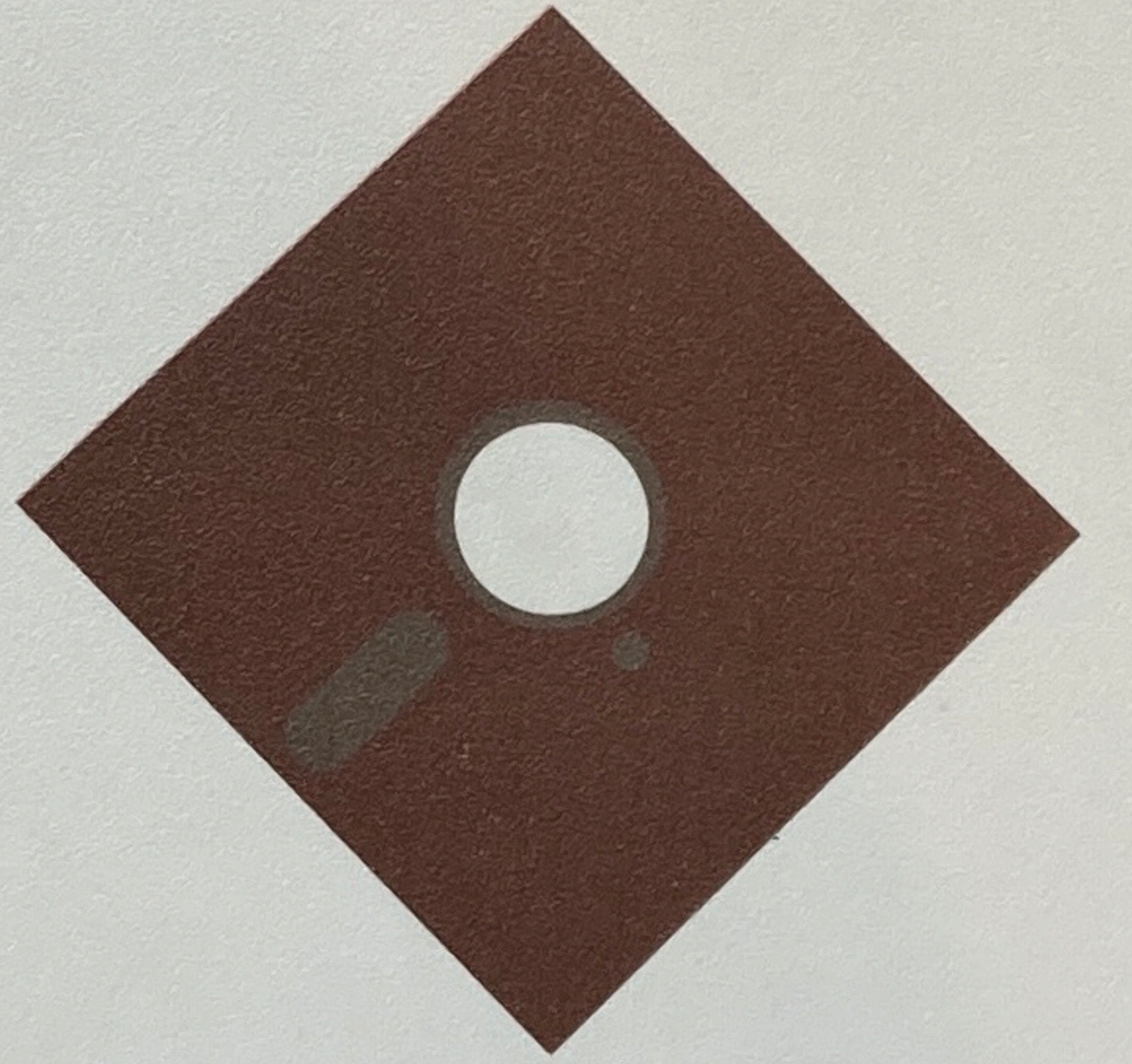
&P   The PAUSE Command

The pause command, &P, causes a program to pause until a key is pressed. The ASCII code of the key used to break the pause is placed in the BASIC integer variable ZK% and removed from the input buffer. At the same time, the number of seconds spent between encountering the &P and pressing a key is placed in the BASIC real variable ZT.

The &P provides an easy way to check for specific single keys. For example, to allow only RETURN to function, use:

```
100  &P: IF ZK% <> 13 THEN 100
```

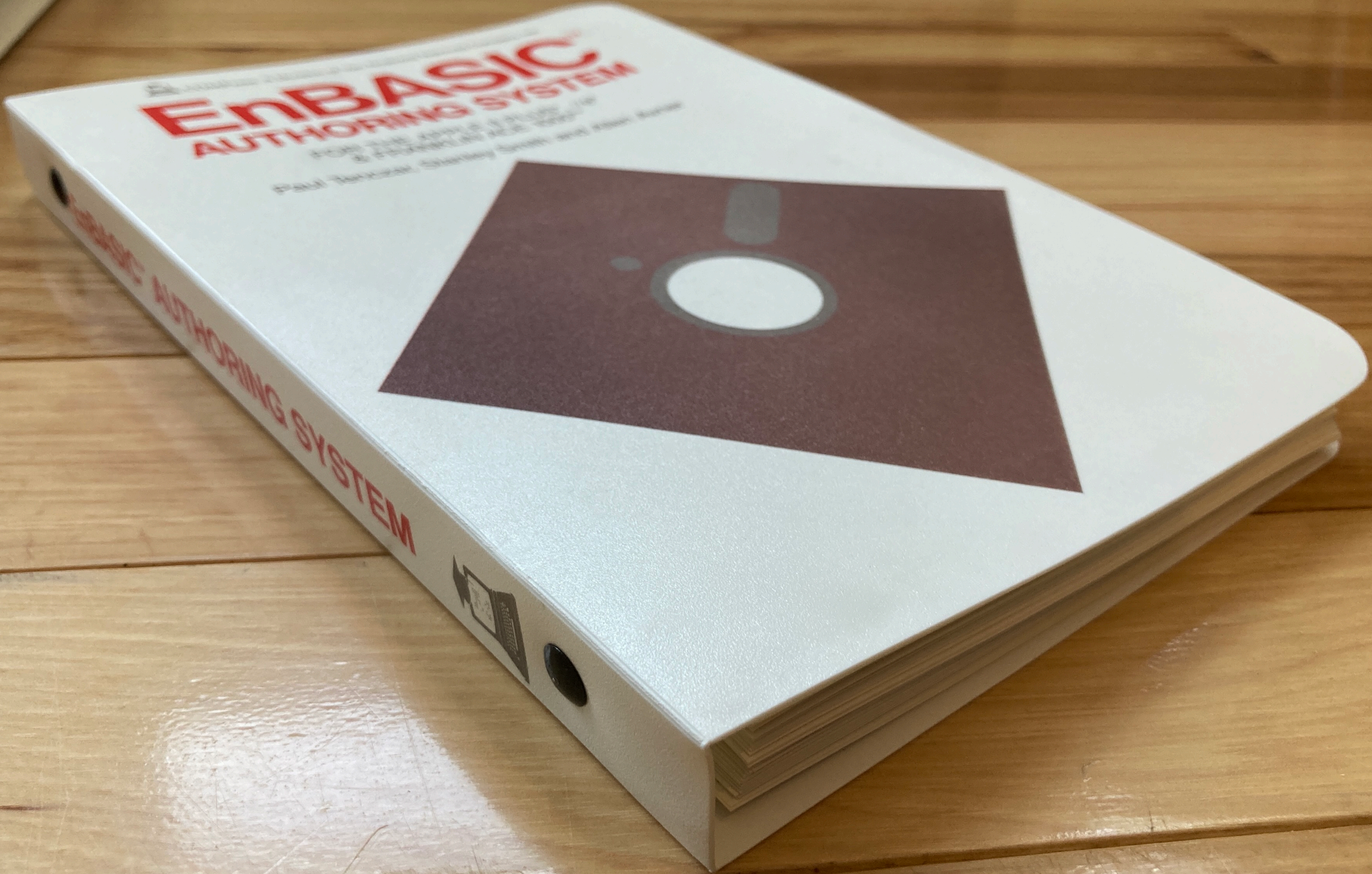
Only if RETURN is pressed does ZK% contain the ASCII code 13. Any other key results in the program looping back to the beginning of line 100. Since ZT is zeroed each time &P



COMPress
A Division of Van Nostrand Reinhold Company, Inc.
P.O. Box 102, Wentworth, NH 03282 (603)764-5831/5225

™ A trademark of Computer Teaching Corporation
* Registered trademark of Apple Computer Company
** Registered trademark of Franklin Computer Corporation

ISBN 0-933694-83-0


EnBASIC
AUTHORING SYSTEM
FOR THE APPLE II PLUS & FRANKLIN ACE 1000
Paul Tenczar, Stanley Smith and Alan Avner
EnBASIC AUTHORING SYSTEM